



Logic for Computer Science. Knowledge Representation and Reasoning.

Lecture Notes
for
Computer Science Students
Faculty EAIIB-IEiT AGH



Antoni Ligęza

Other support material:

<http://home.agh.edu.pl/~ligeza>

[https://ai.ia.agh.edu.pl/pl:dydaktyka:logic:](https://ai.ia.agh.edu.pl/pl:dydaktyka:logic:start#logic_for_computer_science2020)

[start#logic_for_computer_science2020](https://ai.ia.agh.edu.pl/pl:dydaktyka:logic:start#logic_for_computer_science2020)

Basic Ideas behind this Course

Goals of the course:

- to teach selected methods of Logical Knowledge Representation,
- to teach selected methods of Logical Inference,
- with the focus on Propositional Logic Calculus, and also
- First Order Predicate Logic
- with the ultimate sub-goals:
 - building logical models (KR),
 - performing logical reasoning (AR),
 - automated problem solving (APS),
 - analysis of logical properties (LPA).

$$\text{KR} + \text{AR} \longrightarrow \text{APS} + \text{LPA}$$

- to keep the course at understandable basic level and practical applications rather than just theory:
 - necessary background knowledge — but also in an informal way,
 - modern tools — if available,
 - examples + applications,
 - further references; internet sources.
- a reference point: CS157: <http://logic.stanford.edu/classes/cs157/current/>

General Principles

1. **Goal:** to learn some knowledge and practical skills in **logic**.
2. **Syllabus:** see <https://syllabusy.agh.edu.pl/pl/1/1/15/1/4/12/14>
3. **Formal frames:** Regulamin Studiów w AGH¹, but also **logic** and **reasonable, common sense approach**.
4. **Example of particular practical rules GEIST:** <http://geist.agh.edu.pl/pub:teaching:gris>
5. **Forms of work and knowledge acquisition:**
 - lecture,
 - blackboard exercises,
 - *e-learning* (Wikipedia, Coursera,...),
 - **personal, independent study**,
 - discussion, questions and possible answers,
 - consultations.
6. Participation in practical classes is obligatory.
7. Personal attendance and *mindfulness* (**uwaga**) at/during the lectures is **strongly recommended**.
8. A large dose of Enthusiasm, additional activity – supported by personal skills are welcome.
9. **It is strongly recommended** to take personal notes.
10. Everybody builds her/his own **Logical Knowledge Base!**

¹<http://http://www.agh.edu.pl/pracownicy/dokumenty/regulaminy/>

11. Full understanding of the material is of principal importance!.
12. You can ask questions – any time and *almost* on any topics.
13. Tests, exams, practical exercises, etc. — it is obligatory that you present original work on your own.
14. The general principles — according to textitRegulamin Studiów.
15. We have only 7 regular lectures...
16. A schedule/report of lectures: https://ai.ia.agh.edu.pl/pl:dydaktyka:logic:start#logic_for_computer_science2020
17. Only 14 hours...No final exam???

Bibliography

1. **Mordechai Ben-Ari: Mathematical Logic for Computer Science (Logika matematyczna w informatyce). Springer-Verlag, London, 2012 (WN-T, Warszawa, 2005, 2006).**
2. Kenneth A. Ross i Charles R. B. Wright: Discrete Mathematics (Matematyka dyskretna). WN PWN, 2013.
3. Antoni Ligęza: Logical Foundations for Rule-Based Systems. Springer-Verlag, Berlin, 2006. Wydawnictwo AGH, Kraków, 2005.
4. Michael R. Genesereth, Nils J. Nilsson: Logical Foundations of Artificial Intelligence. Morgan Kaufmann Publishers, Inc., Los Altos, California, 1987.
5. Zbigniew Huzar: Elementy logiki dla informatyków. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2007.
6. Stuart Russell, Peter Norvig: Artificial Intelligence. A Modern Approach. Pearson, 2010.
7. Marek Wójcik: Zasada rezolucji. Metoda automatycznego wnioskowania. PWN, Warszawa, 1991.
8. C. L. Chang and R. C. T. Lee: Symbolic Logic and Mechanical Theorem Proving. Academic Press, 1973.
9. Ronald J. Brachman and Hector J. Levesque: Knowledge Representation and Reasoning. Morgan Kaufmann, 2004.
10. Frank van Harmelen, Vladimir Lifschitz, Bruce Porter (Eds.): *Handbook of Knowledge Representation*. Elsevier B.V., Amsterdam, 2008.
<http://ii.fmph.uniba.sk/~sefranek/kri/handbook/>

Support Material: On the Net

Mathematical Logic for Computer Science:

<https://www.weizmann.ac.il/sci-tea/benari/research-activities/mathematical-logic-computer-science-third>

Stanford on-line Course:

<https://www.coursera.org/learn/logic-introduction>

Lectures - On-Line Documentation and Archives:

<https://ai.ia.agh.edu.pl/pl:dydaktyka:logic:start>

1. Wikipedia-pl: http://pl.wikipedia.org/wiki/Logika_matematyczna
2. Wikipedia-en: <http://en.wikipedia.org/wiki/Logic>
3. AI-Lab-Prolog: http://ai.ia.agh.edu.pl/wiki/pl:prolog:prolog_lab
4. EIS-KRR: <http://ai.ia.agh.edu.pl/wiki/pl:dydaktyka:krr:start>
5. ALI-home: home.agh.edu.pl/~ligeza
6. David Poole and Allen Mackworth: Artificial Intelligence. Foundations of Computational Agents. <http://artint.info/>
7. Ulf Nilsson and Jan Maluszynski: Logic, Programming and Prolog. <http://www.ida.liu.se/~ulfni/lpp/>

Introduction to Logic

- Demand for Logic.
- Basic concepts.
- Auxiliary tools.
- Problems and solutions.
- Examples.

Logics - what is that?

Definition 1 *Logika (logos — rozum, słowo, myśl) — nauka o sposobach jasnego i ścisłego formułowania myśli, o regułach poprawnego rozumowania i uzasadniania twierdzeń.*

Definition 2 *Logic (from the Ancient Greek) is the systematic study of the forms of inference, the relations that lead to the acceptance of one proposition, the conclusion, on the basis of a set of other propositions, the premises. More broadly, logic is the analysis and appraisal of arguments.*

Logic = (Axioms) + (Formal Models) + (Valid Inference Rules)

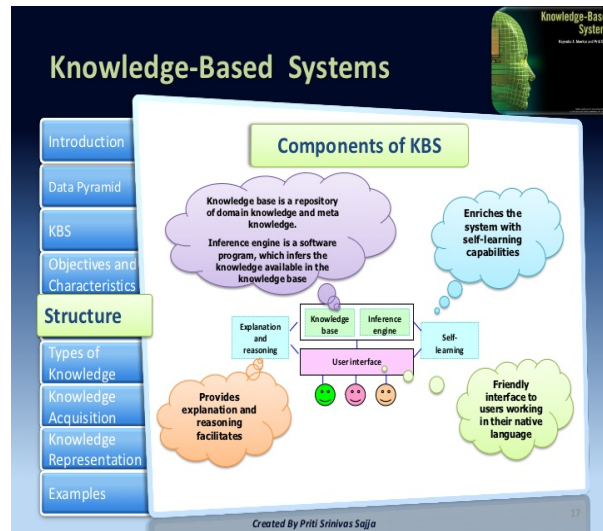
Tools:

- Formal Models – built with Formal Languages:
 - alphabet,
 - syntax,
 - semantics,
 - axioms,
 - inference rules,
 - inference strategies (methods of Theorem Proving);

Supplementary Tools

- Visualization — e.g. Venn's Diagrams,
- Tables (complete specification of cases),
- Trees (better visual presentation),
- Diagrams (graphs; AND-OR graphs; schemes),
- Software tools (e.g. SAT solvers, Prolog, ASP, ATP).

Knowledge-Based Systems: Basic Concepts



We need some tools:

- A formal language for KRR:
 - alphabet,
 - syntax,
 - semantics,
 - inference rules;
- various goals — various types of reasoning,
- inference strategies,
- knowledge acquisition,
- knowledge verification,
- minimal knowledge representation (uniform representation),
- internal knowledge structure (knowledge graphs),
- user interface, explanations, learning, adaptation, optimization.

Przykłady... wieloznaczność języka naturalnego

Żona do męża informatyka

- Kup parówki, a jak będą jajka -- to kup 10.

Mąż w sklepie:

- Są jajka?

- Są.

- To poproszę 10 parówek.

Przykłady wyrażen w języku naturalnym:

- *Mądrej głowie dość po słowie* (Wise head - a word is enough).
 - *Mądrej głowy włos się nie trzyma.*
 - *Dobry kogut to chudy kogut* (A good cock is a skinny cock).
 - *Iloma językami władasz, tylekroć jesteś człowiekiem*
(How many languages do you speak, so many times you are human;
"Wie viele Sprachen du sprichst, sooft mal bist du Mensch.").
- Logika to także język.
- *Historia uczy jednego: nigdy, nikogo, niczego nie nauczyła.*
Czy to zdanie może być prawdziwe?

Logic a tool for Knowledge Representation and Reasoning

Observation:

Not all the systems/processes/... can be modelled with numerical tools.

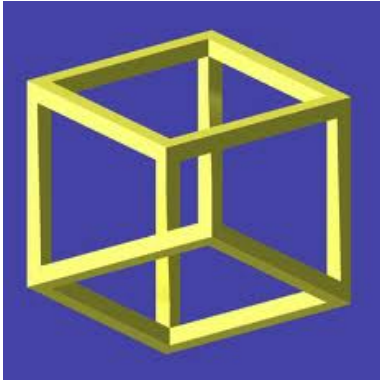
Simultaneously, Natural Language (NL):

- is often inadequate/too poor/...,
- imprecise, incomplete,
- ambiguous– there are different interpretations possible,
- hard for automated processing,
- can lead to inconsistent models/theories,
- is redundant,
- requires the knowledge of *domain ontology*.

Formal language – selected requirements:

- adequate,
- precise,
- unambiguous,
- automatic processing,
- should assure (consistency),
- should allow to produce valid/sound conclusions,
- can assure completeness,
- should be not redundant.

Something has gone wrong...



What one needs logic for?

Po co komu krawat? Sprzedawca krawatów...

- logic allows to formulate definitions of concepts (**marriage, fast tramway, średnioroczny wzrost cen, jakość wykształcenia, jakość kształcenia, jakość procesu kształcenia,...**),
- and the relationship among them — np. taxonomies, mutual relations/roles, ontologies,
- logic introduces order into discussion; rational discussion based on *argumentation*,
- logic provides formal **Knowledge Representation**,
- logic provides valid/correct **inference methods** — deduction,
- but also: **induction** and **abduction**,
- logically specified knowledge can undergo formal analysis:
 - internal; consistency (wewnętrzna niesprzeczność/spójność),
 - completeness (zupełność),
 - minimal representation (minimalna reprezentacja),
 - logical consequence (wynikanie logiczne — logiczna konsekwencja),
 - satisfiability or unsatisfiability (spełnialność lub niespełnialność).
- logic is a formal tool (is widely applied in various domains...).

Where logic is not applied/applicable?? What does not undergo logical rules of reasoning?

Logic – OK – but still some problems...

Natural language:

Paradoks kłamcy:

- *I always lie!* (Eubulides),
- *The Cretans always lie* (Epimenides; he was a Cretan himself),
- Page 1: *The sentence on the other side is true*; Page 2: *The sentence on the other side is false.*
- **Mukator** — Co to jest mukator? To jest coś, co *nie daje się zdefiniować*.

Formal language:

Matematyka:

- Paradox of the set of all sets (Cantor, 1899),
- Russel's paradox (1901): consider $V = \{X : X \notin X\}$. Does it hold that: $V \in V$?
- Barber Paradox: *A barber shaves all residents of the city who do not shave themselves; what should he do with himself?*

Examples of paradoxes: <http://pl.wikipedia.org/wiki/Paradoks>;
<https://en.wikipedia.org/wiki/Paradox>

Formal Knowledge Representation Language: Specification of Requirements

Knowledge Representation Language:

- unique interpretation,
 - expressive power and precision, but
 - automated processing,
 - adequate for the domain,
 - readable for man and machine,
 - extensible,
 - ...
-

Knowledge Base:

- **consistent**, (internal consistency)
- **complete**,
- **valid; sound**, (external consistency)
- **non-redundant**,
- **efficient in problem solving**, (optimal)
- ...

Logic — how it works?

Logic – formal language:

- alphabet,
- *syntax*,
- *semantics*,
- axiomatization

$$\models p \vee \neg p$$

$$\not\models p \wedge \neg p$$

- equivalency-preserving transformations,
- inference rules (logical consequence),
- inference chain – derivation,
- problem, hypothesis, question,
- answer, solution, proof.

System modeling :

- language selection – for adequate modeling,
- model building (inputs – outputs – internal structure),
- model analysis (verification, validation),
- model exploration – theorem proving, SAT,
- model tuning and adaptation; learning.

Example: a simple intro to Propositional Calculus

Alphabet:

- P — set of propositional symbols,
- logical connectives: $\neg, \wedge, \vee, \Rightarrow$,
- parentheses.

Syntax:

- every $p \in P$ is a formula (atomic formula; positive literal),
- if ϕ, ψ are formulas, then:
 - $\neg(\phi)$ is a formula (also: $\neg(\psi)$),
 - $(\phi) \wedge (\psi)$ is a formula,
 - $(\phi) \vee (\psi)$ is a formula,
 - $(\phi) \Rightarrow (\psi)$ is a formula;
 - *no other expression is a formula.*

Each well-formed formula (wff) has a uniquely defined [tree structure](#).

Semantics:

Interpretation — any function I of the form:

$$I: P \rightarrow \{\mathbf{T}, \mathbf{F}\}$$

The definition of interpretation is [extended](#) over the set of all formulae (how to do that ???).

Notation: $I(\phi) = \mathbf{T}$ we shall write $\models_I \phi$; $I(\phi) = \mathbf{F}$ we shall write $\not\models_I \phi$.

For any wff a [truth table](#) can be built.

The symbols \mathbf{T} (*True, true*) and \mathbf{F} (*False, false*) are often replaced with **1** and **0**, respectively.

Rules of inference

https://en.wikipedia.org/wiki/List_of_rules_of_inference: examples + example derivations

Example rules of inference (derivation rules):

Detachment Rule (*Modus Ponens*; Reguła odrywania):

$$\frac{\phi, \phi \Rightarrow \psi}{\psi}$$

Contraposition (*Modus Tollens*; Wnioskowanie przez zaprzeczenie):

$$\frac{\phi \Rightarrow \psi, \neg\psi}{\neg\phi}$$

Disjunctive Syllogism; Disjunction Elimination by Contradiction (Reguła potwierdzania przez zaprzeczenie (wykluczenie)):

$$\frac{\neg\phi, \phi \vee \psi}{\psi}$$

Transitivity Rule (Reguła przechodniości):

$$\frac{\phi \Rightarrow \varphi, \varphi \Rightarrow \psi}{\phi \Rightarrow \psi}$$

Resolution Rule (Reguła rezolucji):

$$\frac{\phi \vee \neg p, p \vee \psi}{\phi \vee \psi}$$

What about the following rules?

Implication + positive conclusion?

$$\frac{\phi \Rightarrow \psi, \psi}{\phi}$$

Implication + negative premise

$$\frac{\neg\phi, \phi \Rightarrow \psi}{\neg\psi}$$

2 Implications + conjunction

$$\frac{p \wedge q, p \Rightarrow r, q \Rightarrow s}{r \wedge s}$$

Proof by Cases: 2 Implications + disjunction

$$\frac{p \vee q, p \Rightarrow r, q \Rightarrow r}{r}$$

Constructive Dilemma: 2 Implications + disjunction

$$\frac{p \vee q, p \Rightarrow r, q \Rightarrow s}{r \vee s}$$

Dual Resolution Rule (Reguła rezolucji dualnej):

$$\frac{\phi \wedge \neg p, p \wedge \psi}{\phi \wedge \psi}$$

Przykład — układ EX-OR

```
% Definicje działania bramek podstawowych
not(1,0).
not(0,1).

and(0,0,0).
and(0,1,0).
and(1,0,0).
and(1,1,1).

or(0,0,0).
or(0,1,1).
or(1,0,1).
or(1,1,1).

% Definicja przykładowego układu - xor

xor(Input1,Input2,Output) :-
not(Input1,N1),
not(Input2,N2),
and(Input1,N2,N3),
and(Input2,N1,N4),
or(N3,N4,Output).
```

Przykład — układ EX-OR

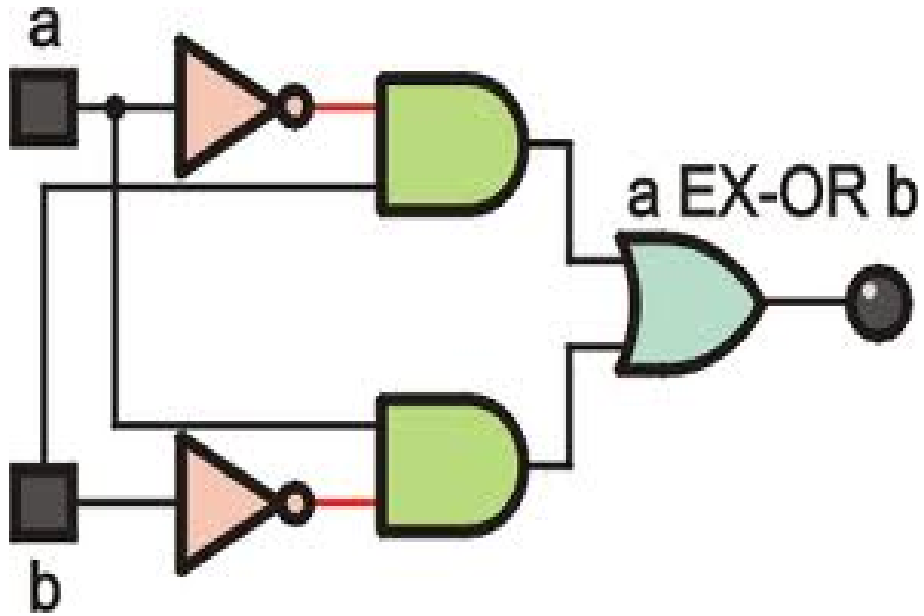
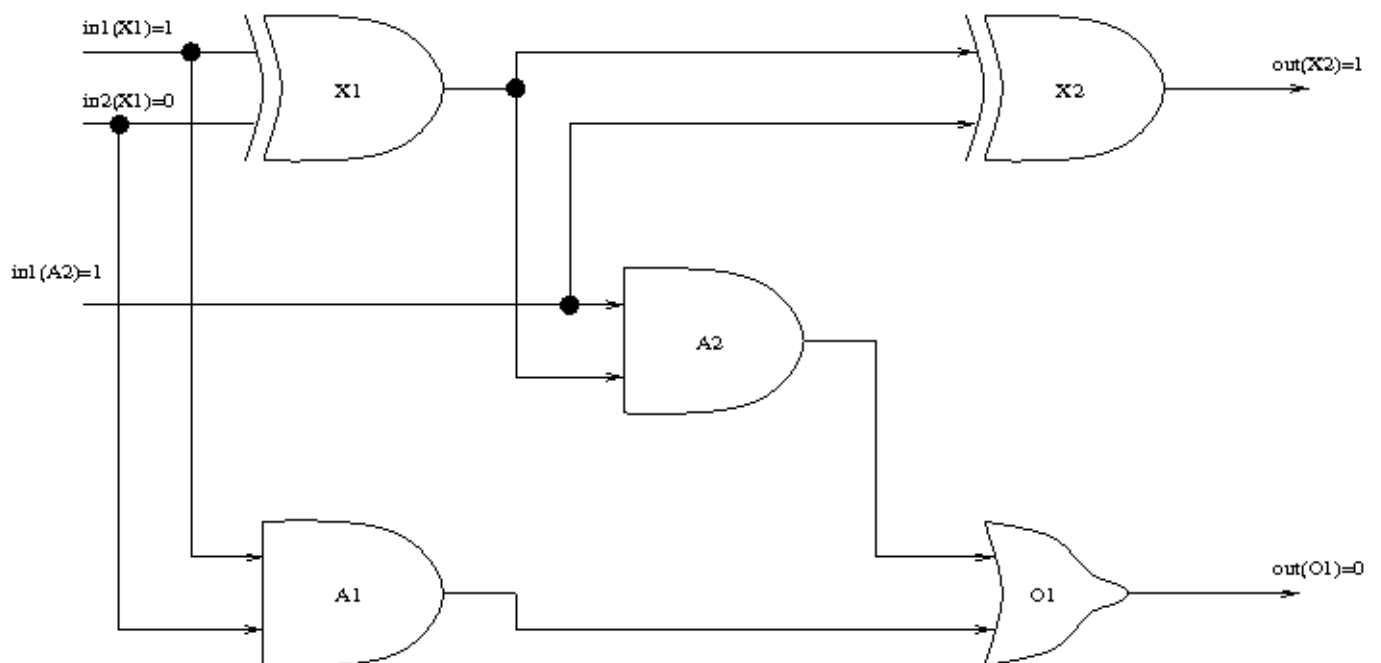


Figure 1: EX-OR digital circuit

Przykład praktyczny

```
foo (I1, I2, I3, O1, O2) :-  
    xor (I1, I2, X1) ,  
    xor (X1, I3, O1) ,  
    and (I1, I2, A1) ,  
    and (X1, I3, A2) ,  
    or (A1, A2, O2) .
```



Logical inferences — examples

Examples of logical inference:

- **deduction** — fundamental logical inference (hypotheses proving);
- SAT: satisfiability checking — model search, constraint analysis;
- detection of inconsistencies; consistency-based reasoning (CBR);
- **tautology** verification;
- **abduction** — making hypotheses (finding out the causes);
- **induction** — generalization;
- case-based reasoning, reasoning by analogy, plausible reasoning; fuzzy reasoning, probabilistic reasoning,...

Examples – applications – and their classification:

- deductive theorem proving (deduction rules, Herbrand, Fitch, forward checking),
- formula for the sum of the infinite geometric sequence (deduction vs. induction),
- SEND + MORE = MONEY,
- diagnosis of multiplier-adder,
- a promise of earnings above average for everyone,
- rational result of power of two irrational numbers,
- polynomial example ($P > 0$ and $\forall \epsilon > 0: P < \epsilon$),
- practical examples:

- 7-segment display - for 16-bar code
- 3/5 voting system
- lock-2-doors to the bank,
- railway crossing,

Logic for KRR – Tasks and Tools

- Theorem Proving – Verification of Logical Consequence:

$$\Delta \models H;$$

- Automated Inference – Derivation:

$$\Delta \vdash H;$$

- SAT (checking for models) – satisfiability:

$$\models_I H;$$

- un-SAT verification – unsatisfiability:

$$\not\models_I H \quad \text{for any interpretation } I;$$

- Tautology verification (completeness):

$$\models H$$

- valid inference rules – checking:

$$(\Delta \vdash H) \longrightarrow (\Delta \models H)$$

- complete inference rules – checking:

$$(\Delta \models H) \longrightarrow (\Delta \vdash H)$$

Inference example

A – signal from process,

P – signal added to a queue,

B – signal blocked by process,

D – signal received by process,

S – state of the process saved,

M – signal mask read,

H – signal management procedure activated,

N – procedure executed in normal mode,

R – process restart from context,

I – process must re-create context.

Rules — axiomatization:

$A \longrightarrow P,$

$P \wedge \neg B \longrightarrow D,$

$D \longrightarrow S \wedge M \wedge H,$

$H \wedge N \longrightarrow R,$

$H \wedge \neg R \longrightarrow I,$

Facts:

$A, \neg B, \neg R.$ What can be inferred from the facts? What are the logical consequences of the KB? Which facts are True and which ones are False?

Is the system complete?

A Problem to Consider: Tracking the Murderer

Some knowledge specification — in natural language:

- If Sarah was drunk then either James is the murderer or Sarah lies,
- Either James is the murderer or Sarah was not drunk and the crime took place after midnight,
- If the crime took place after midnight then either James is the murderer or Sarah lies,
- Sarah does not lie when sober.

Introduction symbols and transformation to formal specification:

- A = James is the murderer,
- B = Sarah is drunk,
- C = Sarah lies,
- D = The murder took place after midnight.

$$B \implies A \vee C$$

$$A \vee (\neg B \wedge D)$$

$$D \implies (A \vee C)$$

$$C \implies B$$

Questions:

Who is the murderer? Which facts are true/false? Is the system consistent? How many models does it have (if consistent)? What are the exact models?

Example: Unicorn



Given the following Knowledge Base (KB):

- If the unicorn is mythical, then it is immortal
- If the unicorn is not mythical, then it is a mortal mammal
- If the unicorn is either immortal or a mammal, then it is horned
- The unicorn is magical if it is horned

answer the following questions:

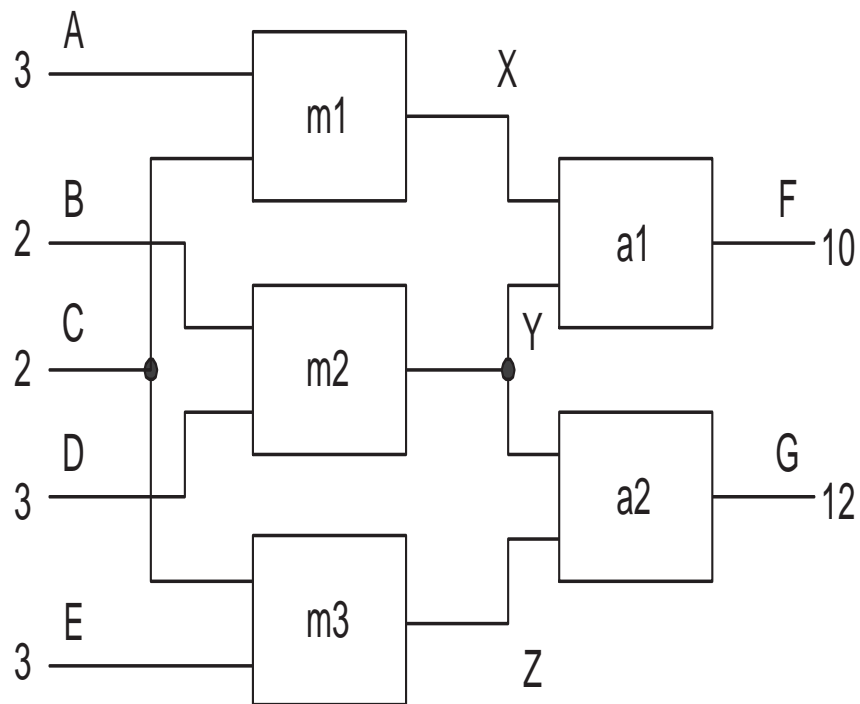
- Is the unicorn mythical? (M)
- Is it magical? (G)
- Is it horned? (H)

In terms of logic:

$$\text{KB} \models G, H, M$$

$$\text{KB} \vdash G, H, M$$

Abductive Inference and Consistency-Based Reasoning



Questions:

- Does the system work OK (Fault Detection)?
- Which component(s) is(are) faulty (Fault Isolation)?
- What are the diagnoses (minimal diagnoses)?